

Laborator 3

Acest laborator este dedicat parcurgerii principiilor API-urilor RESTful

[Documentul laboratorului](#)

[Diferenta intre URI si URL](#)

Constrangerile API-urilor RESTful:

1. [Uniform interface](#)

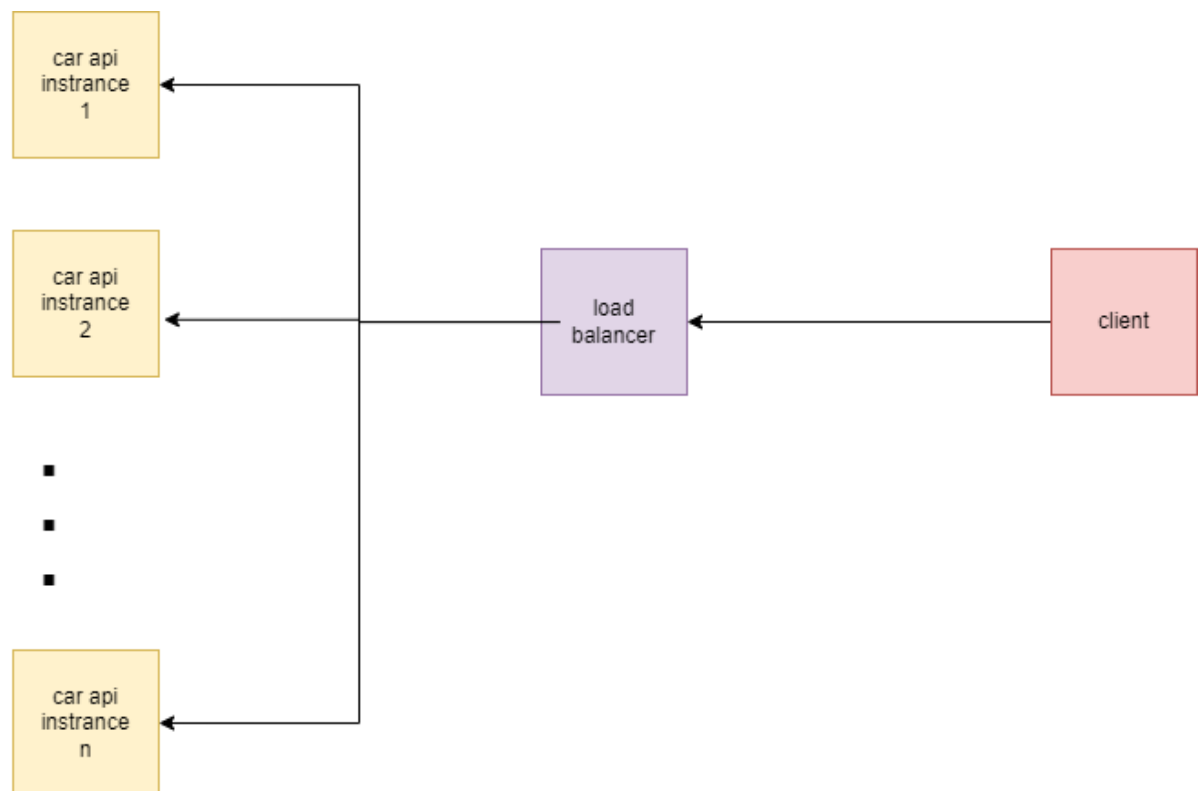
Exemplu API care implementeaza HATEOAS in [acest repository](#)

2. [Client-server](#)

3. [Stateless](#)

4. [Cacheable](#)

5. [Layered system](#)



6. [Code on demand](#)

Denumirea API-urilor REST

Intelegerea codurilor de stare HTTP

Colectii de cereri in POSTMAN

Tema 2, cu predare in saptamana 4 (16.03.2022)

Create an application that provides a RESTful API. Observation : Is mandatory to use at least: GET, POST, PUT, DELETE. It is very important to respect all additional requirements specified in the laboratory.

Additional Information:

- You are not allowed to use frameworks for this homework.
- Your API should communicate using either JSON or XML.
- You should provide two routes for each HTTP verb (ex. GET can be applied either on a resource or on a collection of resources - please refer to the table in section Using HTTP Methods for RESTful Services).
- Its uniform interface should be “Resource-based” (refer to What is REST? section).
- Make sure you design your API to be intuitive and easy to use by naming your resources in a logical manner (refer to Resource Naming section)
- Make sure to respect the idempotence and safeness properties for the HTTP verbs that have one or both (refer to Idempotence of REST APIs)
- Make sure your API responds with the corresponding status codes for each situation it reaches in the processing of a request (ex. 200 - Ok, 404 - Not Found, 500 - Internal Server Error etc.)
- For storing the resources, you must use a persistent storage (XML file or any database of your choice)
- You have to create a POSTMAN (<https://www.getpostman.com/>) collection of requests for each call that can be made in your API, export it and commit it on your repository alongside with your code. Students that have an API without a prepared collection of requests will NOT be evaluated!

General observations:

- The homework can be implemented in any programming language.
- This homework should be solved individually by each of you.
- This homework, like the previous one, shall be presented using the video projector, so make sure to bring all the tools/adapters you may need.
- The complexity of the API will also be an evaluation criteria in the final homework grade.